

# Raport Rozproszone Systemy Informatyczne

## WCF – baza danych użytkowników

Aleksandra Wolska  
Szymon Łopuszyński  
Konfiguracja dwumaszynowa

1. Wymagania
  - a. Visual Studio 2022 z pakietem roboczym tworzenia aplikacji ASP.NET i aplikacji internetowych, w tym opcjonalny pakiet “Windows Community Foundation”
  - b. Dwie maszyny połączone znajdujące się w tej samej sieci udostępnianej przez hotspot
  - c. Import autorskiej biblioteki MyData do wyświetlania danych
2. Definiowanie biblioteki WCF z kontraktem
  - a. Tworzymy nowy projekt WCFService
  - b. Wybieramy szablon “Visual C# WCF Service Library”
  - c. W dalszej konfiguracji wybieramy wersję platformy .NET 4.8
  - d. Tworzymy plik IDatabaseService.cs, w którym definiujemy kontrakt serwisu, oraz kontraktowy typ danych User naszego użytkownika.
  - e. Dodajemy anotację `[ServiceKnownType(typeof(User))]` przed deklaracją kontraktu, aby umożliwić przesyłanie obiektów typu User przez nasz serwis.
3. Implementacja kontraktu
  - a. Utworzenie nowej klasy DatabaseService implementującej interfejs IDatabaseService.cs
  - b. W pliku DatabaseService.cs definiujemy metody z interfejsu do obsługi kolekcji użytkowników, oraz pole klasy typu ArrayList, w której będziemy przechowywać dane użytkowników.
  - c. W celu obsługi wywołań asynchronicznych do klasy implementującej kontrakt serwisu dodajemy anotację  
`[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single, ConcurrencyMode = ConcurrencyMode.Multiple)]`
  - d. Implementacja zgłaszania wyjątków przez serwer opakowywując je w generyczną klasę `FaultException<>`
4. Hostowanie usługi
  - a. Tworzymy aplikację konsolową .Net w Visual Studio
  - b. W Solution Explorer w folderze References dodajemy referencję do System.ServiceModel oraz do naszego projektu serwisu WCFService
  - c. W pliku Program.cs tworzymy adres URI naszego hostowanego serwisu, deklarujemy bindingi oraz endpointy, które host będzie obsługiwał.
    - i. Wykorzystano bindingi BasicHttpBinding oraz WSHttpBinding

- ii. Otwieramy port 10000 w firewallu systemowym aby umożliwić komunikację pomiędzy maszynami

```
MyData.MyData.info();
Uri baseAddress = new Uri("http://192.168.43.18:10000/DatabaseService");
ServiceHost myHost = new ServiceHost(typeof(DatabaseService), baseAddress);

BasicHttpBinding myBinding = new BasicHttpBinding();
ServiceEndpoint endpoint1 = myHost.AddServiceEndpoint(typeof(IDatabaseService), myBinding,
"endpoint1");
WSHttpBinding binding2 = new WSHttpBinding();
binding2.Security.Mode = SecurityMode.None;
ServiceEndpoint endpoint2 = myHost.AddServiceEndpoint(typeof(IDatabaseService), binding2,
"endpoint2");
ServiceMetadataBehavior smb = new ServiceMetadataBehavior();
smb.HttpGetEnabled = true;
myHost.Description.Behaviors.Add(smb);
```

## 5. Implementacja usługi klienta

### a. Główne kroki

- i. Tworzymy nową solucję z szablonu C# Console App (.NetFramework) w wersji 4.8.
- ii. Dodajemy referencję na System.ServiceModel
- iii. Definiujemy plik IDatabaseService.cs, który jest kopią kontraktu serwisu z projektu WcfService. Posiada on poniższe metody

```
[OperationContract]
ArrayList getAllUsers();
```

```
[OperationContract]
Task<ArrayList> getAllUsersSleep();
```

```
[OperationContract]
int getUserDatabaseSize();
```

```
[OperationContract]
User addUser(User user);
```

```
[OperationContract]
User updateUser(User user);
```

```
[OperationContract]
User deleteUser(int id);
```

### b. W pliku Program.cs:

- i. tworzymy instancję klienta DatabaseClient, który jest dodatkową klasą utworzoną w celu obsługi udostępnionych usług serwisu i komunikacji z użytkownikiem.

```
public DatabaseClient(string binding = "WSHttpBinding_IDatabaseService")
{
```

```

        client = new DatabaseServiceClient(binding);
    }

```

### c. Utworzenie proxy klienta

- i. Uruchamiamy hosta naszej usługi
- ii. Dodajemy do klienta referencji serwisową do zdefiniowanej usługi podając adres usługi
- iii. Plik AppConfig zawiera dostosowane do potrzeb aplikacji, oraz ustawienie timeoutu po stronie klienta. Flaga "sendTimeout" ustawiona na 5 sekund, oznacza oczekiwanie na odpowiedź przez 5 sekund od momentu zainicjowania wysłania wiadomości, do zakończenia otrzymywania odpowiedzi z hosta.

#### 1. bindingi:

```

<bindings>
  <basicHttpBinding>
    <binding name="BasicHttpBinding_IDatabaseService" />
  </basicHttpBinding>

  <wsHttpBinding>
    <binding name="WSHttpBinding_IDatabaseService"
              sendTimeout="00:00:05">
      <security mode="None" />
    </binding>
  </wsHttpBinding>
</bindings>

```

#### 2. endpointy,

```

<client>
  <endpoint address="http://192.168.43.18:10000/DatabaseService/endpoint1"
            binding="basicHttpBinding"
            bindingConfiguration="BasicHttpBinding_IDatabaseService"
            contract="ServiceReference1.IDatabaseService"
            name="BasicHttpBinding_IDatabaseService" />
  <endpoint address="http://192.168.43.18:10000/DatabaseService/endpoint2"
            binding="wsHttpBinding" bindingConfiguration="WSHttpBinding_IDatabaseService"
            contract="ServiceReference1.IDatabaseService"
            name="WSHttpBinding_IDatabaseService" />
</client>

```

- d. Implementacja obsługi zgłaszanych na serwerze wyjątków opakowanych w generyczną klasę `FaultException<>`
- e. Wywołujemy operacje serwisu z użyciem klienta proxy
- f. Zamykamy klienta

## 6. Działanie programu

Klient: Wylistowanie i usunięcie  
użytkownika

C:\Users\Aleksandra\Desktop\Polibuda\RSI\REPO\RSI...

```
Aleksandra Wolska, 251810
Szymon Łopuszyński, 260454
25 kwietnia, 09:49:46
4.0.30319.42000
Aleksandra
Microsoft Windows NT 6.2.9200.0
192.168.43.194
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
1
```

```
Id: 0 Aleksandra Wolska, age: 22
Id: 1 Szymon Łopuszyński, age: 22
```

```
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
5
```

```
Podaj id użytkownika:
0
```

```
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
1
```

```
Id: 1 Szymon Łopuszyński, age: 22
Menu:
```

Klient: Edycja uzytkownika

```
Menu:
1. Pokaż wyszyskich uzytkowników
2. Wybierz uzytkownika
3. Dodaj uzytkownika
4. Edytuj uzytkownika
5. Usuń uzytkownika
6. Pokaż wszystkich uzytkowników async
0. Exit
4
Podaj id uzytkownika:
1
Podaj imie i nazwisko uzytkownika:
Szymek Szymański
Podaj wiek uzytkownika: 21
Menu:
1. Pokaż wyszyskich uzytkowników
2. Wybierz uzytkownika
3. Dodaj uzytkownika
4. Edytuj uzytkownika
5. Usuń uzytkownika
6. Pokaż wszystkich uzytkowników async
0. Exit
1
Id: 1 Szymek Szymański, age: 21
Menu:
```

Klient: Dodanie uzytkownika

```
Menu:
1. Pokaż wyszyskich uzytkowników
2. Wybierz uzytkownika
3. Dodaj uzytkownika
4. Edytuj uzytkownika
5. Usuń uzytkownika
6. Pokaż wszystkich uzytkowników async
0. Exit
3
Podaj id uzytkownika:
3
Podaj imie i nazwisko uzytkownika:
Kasia Kowalska
Podaj wiek uzytkownika: 23
Menu:
1. Pokaż wyszyskich uzytkowników
2. Wybierz uzytkownika
3. Dodaj uzytkownika
4. Edytuj uzytkownika
5. Usuń uzytkownika
6. Pokaż wszystkich uzytkowników async
0. Exit
1
Id: 1 Szymek Szymański, age: 21
Id: 3 Kasia Kowalska, age: 23
Menu:
```

Klient: Dodanie użytkownika o kluczu istniejącym w bazie

```
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
3
Podaj id użytkownika:
3
Podaj imie i nazwisko użytkownika:
Ala Alowska
Podaj wiek użytkownika: 25
User already exists in the database
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
1

Id: 1 Szymek Szymański, age: 21
Id: 3 Kasia Kowalska, age: 23
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
```

Klient: Pokazanie użytkownika

```
Id: 1 Szymek Szymański, age: 21
Id: 3 Kasia Kowalska, age: 23
Menu:
1. Pokaż wszystkich użytkowników
2. Wybierz użytkownika
3. Dodaj użytkownika
4. Edytuj użytkownika
5. Usuń użytkownika
6. Pokaż wszystkich użytkowników async
0. Exit
2
Podaj id użytkownika
3
Id: 3 Kasia Kowalska, age: 23
Menu:
```

Klient: Operacja asynchroniczna	<pre>Menu: 1. Pokaż wszystkich użytkowników 2. Wybierz użytkownika 3. Dodaj użytkownika 4. Edytuj użytkownika 5. Usuń użytkownika 6. Pokaż wszystkich użytkowników async 0. Exit 6  Menu: 1. Pokaż wszystkich użytkowników 2. Wybierz użytkownika 3. Dodaj użytkownika 4. Edytuj użytkownika 5. Usuń użytkownika 6. Pokaż wszystkich użytkowników async 0. Exit Operacja zajela zbyt duzo czasu</pre>
Zrzut ekranu z hosta serwisu	<pre>Aleksandra Wolska, 251810 Szymon Topuszyński, 260454 25 kwietnia, 09:47:55 4.0.30319.42000 shibe Microsoft Windows NT 6.2.9200.0 192.168.43.18 00:00:30 ---&gt; Endpointy: Service endpoint: BasicHttpBinding_IDatabaseService Binding: System.ServiceModel.BasicHttpBinding ListenUri: http://192.168.43.18:10000/DatabaseService/endpoint1 Service endpoint: WSHHttpBinding_IDatabaseService Binding: System.ServiceModel.WSHHttpBinding ListenUri: http://192.168.43.18:10000/DatabaseService/endpoint2 Service is started and running. Press &lt;ENTER&gt; to STOP service...  All users Remove user Id: 0 Aleksandra Wolska, age: 22 All users Update user Id: 1 Szymek Szymański, age: 21 All users Add user Id: 3 Kasia Kowalska, age: 23 All users All user async start All user async end All users All users Add user Id: 3 Ala Alowska, age: 25 All users All user async start All user async end</pre>

Źródła:

- Instrukcja Dr. Frasia - Cwiczenie 3 (wer.b - dodany jeden komentarz)
- <https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/configuring-timeout-values-on-a-binding>
- <https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/configuring-timeout-values-on-a-binding>
- <https://learn.microsoft.com/en-us/dotnet/framework/wcf/wcf-error-handling>